

## Examen - Méthodes formelles - durée 2h

Mars 2006 - documents du cours autorisés

Pour vous évaluer, nous allons privilégier la qualité de vos réponses à leur quantité.

### Exercice 1

On considère des polygones qui décrivent des figures géométriques : triangle, carré, rectangle, etc. On veut développer une application logicielle pour construire des images avec des polygones puis étudier et manipuler ces images.

Pour ce faire, on fait une modélisation B où on caractérise chaque image par une collection de figures géométriques ayant chacune une position. Une position est donnée ici par deux valeurs entières (x, y). Chaque figure est ainsi caractérisée par sa forme (triangle, carré, rectangle, etc), sa couleur (rouge, vert, bleu, etc) et sa position.

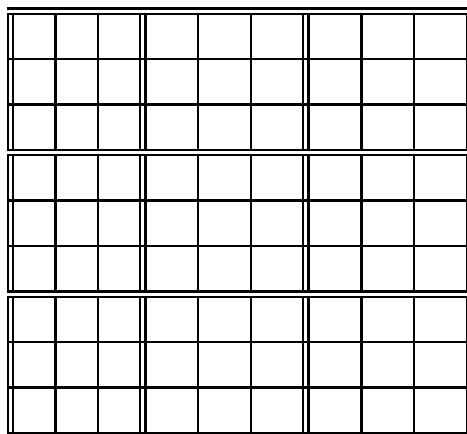
L'application dispose d'une corbeille qui peut contenir des figures (copiées ou coupées comme on le verra ci-après). L'application dispose aussi d'une figure courante, c'est celle qui est sélectionnée.

On veut spécifier en B les opérations suivantes :

- **CreerFigure**. Cette opération attend comme paramètre le type de la figure à créer. Une couleur par défaut est alors donnée à la figure créée. Une position par défaut est donnée à la figure créée. L'opération renvoie la figure créée.
  - **ModifierCouleur**. Elle attend une figure et une couleur en paramètre. L'opération modifie la couleur de la figure donnée en paramètre.
  - **CreerImage**. Pas de paramètre. Elle crée une image qui n'a pour le moment aucune figure. L'opération renvoie l'image créée.
  - **AjouterFigure**. Cette opération attend deux paramètres : une image et une figure. L'opération ajoute la figure à l'image.
  - **ModifierPosition**. Nécessite trois paramètres, une figure et deux valeurs définissant une position. La position de la figure est modifiée.
  - **Selectionner**. Cette opération attend deux paramètres : x et y qui sont des valeurs entières. L'opération sélectionne la figure dont les valeurs de position sont identiques à x, y. Il y a au plus une figure sélectionnée à tout moment. Une sélection annule la sélection précédente.
  - **CouperFigure**. Cette opération n'attend aucun paramètre. Elle coupe la figure actuellement sélectionnée ; la figure ainsi coupée va dans la corbeille.
  - **CopierFigure**. Elle copie la figure actuellement sélectionnée, la copie va dans la corbeille.
1. Proposez une spécification B qui modélise l'espace d'état de l'application (restreinte aux hypothèses énoncées ci-dessous).
  2. Spécifiez les opérations présentées ci-dessus. Complétez au besoin les hypothèses de travail.
  3. Donnez et prouvez l'obligation de preuve de cohérence de 2 opérations.

## Exercice 2

On veut écrire une application qui analyse des grilles de Sudoku<sup>1</sup>. Dans cet exercice on fait les hypothèses suivantes.



Une grille sudoku a 3 lignes et 3 colonnes contenant des matrices. La grille est ainsi constituée de 9 matrices carrée 3 x 3. Les matrices contiennent des valeurs entières entre 1 et 9.

Un joueur de sudoku part d'une grille partiellement remplie. Le jeu consiste alors à remplir complètement la grille de façon à ce que chaque matrice soit correcte et la grille complète soit correcte.

Une **matrice est dite correcte** si les valeurs qui y sont contenues sont comprises entre 1 et 9 et aucune valeur n'est présente plus d'une fois. On considère qu'il n'y a pas de trou dans la matrice.

Une **grille sudoku est correcte** si toutes ces matrices sont correctes et si chaque ligne (resp. chaque colonne) de la grille contient des valeurs entre 1 et 9 et aucune valeur n'est présente plus d'une fois.

On veut construire une machine abstraite B avec des opérations (ou plusieurs machines avec des opérations) pour analyser la grille proposée par un joueur.

1. Spécifiez en B une matrice (3 x 3) sudoku.
2. Spécifiez en B une grille sudoku.
3. Spécifiez en B une opération qui analyse la correction d'une matrice (remplie).
4. Spécifier en B une opération qui analyse la correction de toutes les matrices d'une grille (remplie).

Bonus ...

5. Spécifiez l'analyse de correction d'une ligne de la grille.

---

<sup>1</sup>jeu de réflexion d'origine japonaise